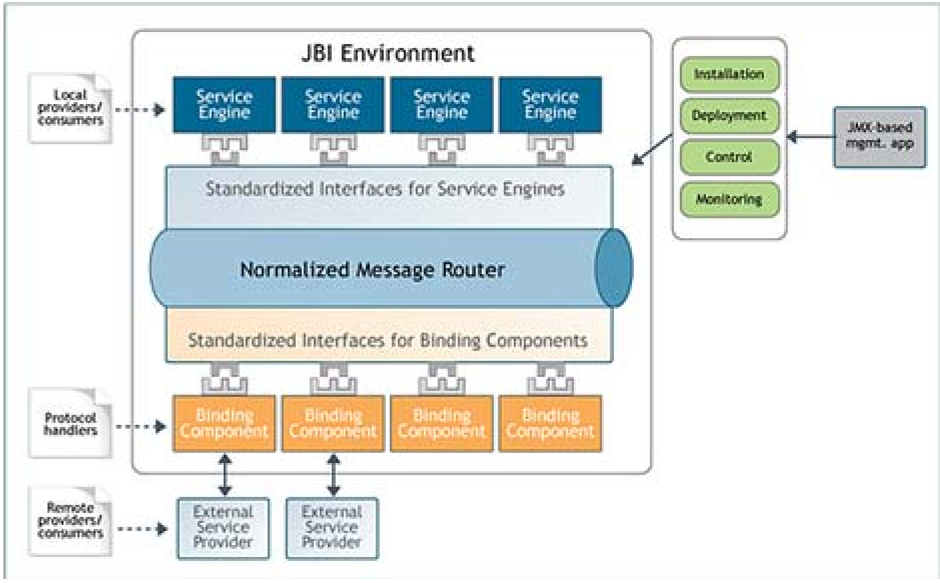
FUSE includes four open source SOA components: **FUSE ESB, FUSE Message Broker, FUSE Services Framework and FUSE Mediation Router**. The four components are combined with development and management tools to create a comprehensive enterprise integration infrastructure and are deployed in many large, mission-critical IT applications.

Enterprise service buses are key enablers for agile and effective integration across enterprise systems. **FUSE ESB** is a JBI compliant Enterprise Service Bus based on Apache Servicemix. It is lightweight, standards-based, and performs the common ESB functionality such as transport mediation, intelligent routing, and message transformation and adds true location transparency.   It has advanced features such as clustering, hot deployment, multiple deployment options and many out the box components/adapters.

The architecture of FUSE ESB 3.x is based on Java Business Integration (JBI) specification (JSR-208) which defines an integration architecture based on service oriented concepts. Applications are broken up into decoupled functional units. The functional units are deployed into JBI components that are hosted within the JBI environment. The JBI environment provides message normalization and message mediation between the JBI components and is as pictured below.


JBI Environment
Local providers/ consumers
Service Engine
Service Engine
Service Engine
Service Engine
Standardized Interfaces for Service Engines
Normalized Message Router
Standardized Interfaces for Binding Components
Protocol handlers
Binding Component
Binding Component
Binding Component
Binding Component
Remote providers/ consumers
External Service Provider
External Service Provider
Installation
Deployment
Control
Monitoring
JMX-based mgmt. app

In addition FUSE ESB Release 4.1 adds OSGi functionality. The OSGi framework provides a fast and standard way to create, deploy and provision integration components and manage concurrent versions of software components and dependencies. OSGi also allows applications to be updated without downtime. Components can be dynamically added to a running system and existing running components can be stopped and updated without taking down the entire application.

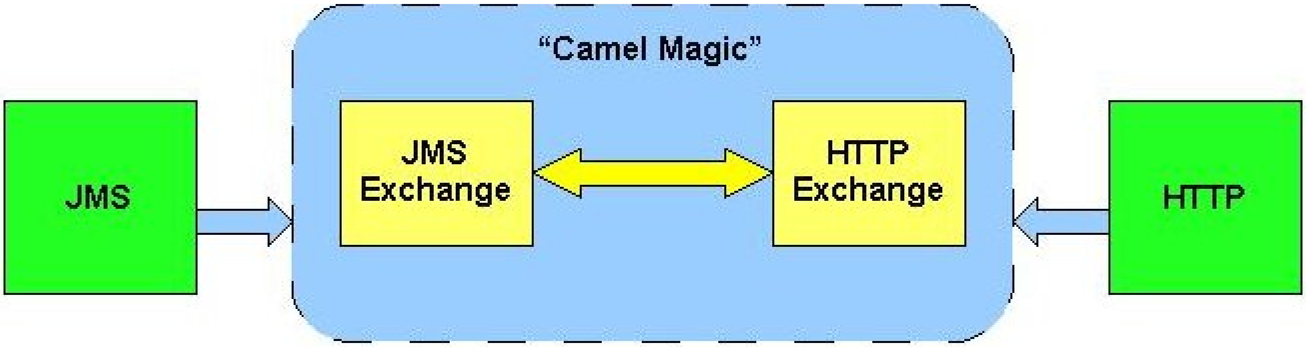
Fuse Services Framework is a product aimed at simplifying the development and deployment of Web Service-based applications, both at the client (consumer) and server/service (provider) side. It is based on Apache CXF and enables developers and architects to easily create Web Services and/or invoke Web Services from within the Java-based development environment. It is the lightweight foundation of a distributed SOA infrastructure and a key component in FUSE ESB.

The Framework consists of both a set of library files (Java Archive files) which implement the run-time Web Services Stack (http, SOAP, WSDL, UDDI) and facilities. Services Framework supports both contract-driven (i.e. from WSDL files) and code-first (i.e. from Java files) styles of development with generators for both Java-to-WSDL and WSDL-to-Java.

Out of the box, there is support for HTTP and JMS transports, but Services Framework allows developers to add custom bindings fairly easily. In addition, there is basic support for SOAP and "raw XML" as bindings, but again extendible by developers.

FUSE Mediation Router is based on Apache Camel which is a lightweight, rule-based routing and process mediation engine. FUSE MR implements the Enterprise Integration Patterns (EIPs), as defined in Gregor Hohpe and Bobby Woolf's book, through a dynamic scripting language or graphical editor/debugger.

Mediation Router, by default, will do the best effort to convert from one protocol to another. It does this by filling in all necessary protocol values (such as headers, etc.) and by converting one exchange object into another.



In this simple way data can be moved from a file to a JMS queue for example. In more complex scenarios best effort may not work and it is the responsibility of the developer to make sure conversion happens properly. In this case Mediation Router provides the object to be populated.

FUSE Message Broker is an open source JMS message broker that is based on Apache ActiveMQ. Message Broker is the JMS platform of choice for scalable, high-performance infrastructure to connect processes across heterogeneous systems. Message Broker is JMS 1.1 compliant allowing application components to create, send, receive, and read messages for reliable, loosely coupled communication.

Message Broker delivers large amounts of data efficiently and reliably and provides "Enterprise Features" such as clustering and failover to ensure high availability. It supports queue and topic messaging domains, a variety of persistent and non-persistent messaging, local JMS transactions, and XA-enabled JMS API. It can use many databases for JMS persistence in addition to cache and journal persistency. Asynchronous messaging, publish/subscribe, point-to-point and stream-based messaging are supported with connections that are durable, fault tolerant and high availability.